

RTCA Special Committee 186, Working Group 5

ADS-B UAT MOPS

Meeting #4

**Preliminary Report on
UAT Synchronization Issues**

**Prepared by Warren J. Wilson
The MITRE Corp.**

SUMMARY

In partial fulfillment of Action Item 3-23, this paper addresses some issues related to the synchronization process for UAT ADS-B messages.

1. Introduction

This paper examines some issues related to the UAT burst synchronization process. The primary purpose of any synchronization process (including that of UAT) is to detect the presence of a signal and to determine as accurately as possible the correct timing of the signal so that it can be successfully demodulated.

For UAT, synchronization is supported by a 36-bit sequence that occurs at the beginning of every burst. A simplified version of the synchronization process might consist of comparing samples of the incoming bit stream with the expected sequence. If more than a certain number (say, 31) of these bits were correct, a valid signal would be considered detected. The length of the sequence, the particular sequence used, and the threshold are chosen so that there is a reasonably low probability of false alarm (“synchronizing” with no valid sequence actually present or with the timing incorrect by more than one bit period). In order to insure that the synchronization process determines the bit sampling time to within a small fraction of a bit period (to optimize performance), the waveform is typically sampled more than once per bit during the synchronization process. In the discussion below, we will assume that the sampling rate is six times the bit rate (i.e., 6.25 Msps).

Synchronization for the UAT ADS-B messages is complicated by the fact that they are transmitted pseudorandomly by each user and there is, therefore, a high probability of signal overlap. Overlap is not necessarily an overwhelming problem because UAT is fairly robust in the presence of self-interference. This is primarily due to the fact that the waveform is basically binary FM. Very good performance results when the desired-to-undesired ratio (D/U) is as little as 6 dB (with a single interferer). However, this property could be rendered irrelevant if receivers which have already synchronized to weak signals cannot resynchronize (or “retrigger”) on strong signals arriving slightly later. Thus, good performance in the presence of a heavy self-interference environment is dependent upon the ability to continue to search for synchronization while demodulating a signal. If a new synchronization correlation is found during the course of demodulating a burst, the receiver can either switch from processing the old signal to the new one, or it can attempt to demodulate both. There is some danger in the switching option because the sequence used for synchronization (or something very close to it) may be embedded in a valid message. In the next section we will show an example of how to implement the option of demodulating both.

2. Synchronization Process Description

Figure 1 shows a top-level diagram of a possible UAT synchronization scheme. It shows that after the received signal passes through an IF filter, it is detected by a limiter/discriminator that generates a string of ones and zeros at a rate of 6 times per symbol (i.e., 6.25 Msps). The sample stream then goes through a correlator, which is looking for the synchronization pattern. As the samples stream out of the correlator, they enter a 6 by N byte memory. This is arranged so that each “row” of the memory contains samples separated by the bit period. In the meantime whenever the box labeled

“threshold” determines that a signal is present, it will attach a flag to the appropriate location in the memory. The processor will then read out bits from the memory (on a row-by-row basis) according to the locations of the flags. This read-out process will allow the processor to deal with overlapping messages, whether or not they are in the same row of the memory.

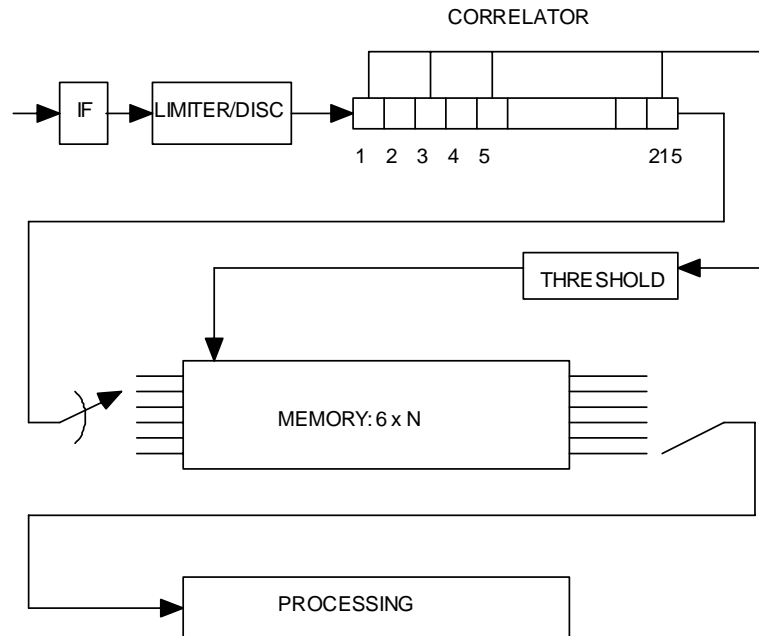


Figure 1: UAT Receiver Processing

Note that *all* samples follow this path, even when “information” is being processed. Thus, the receiver can continuously look for new synchronization patterns. The “depth” of this process (the worst-case minimum of the number of overlapping signals that can be accommodated) depends on the size of the memory. If each long ADS-B message is about 50 bytes long, then a depth of 4 would require $6 \times 50 \times 4 = 1200$ bytes, which is tiny. Depth is not a problem for the memory.

In the paragraphs below, some of the components of figure 1 are dealt with in a little more detail.

The **correlator** is basically a tapped delay line whose length is long enough to hold about $36 \times 6 = 216$ samples. The taps are arranged so that every other location is used to compare the incoming sample sequence with the fixed synchronization sequence. The expected sequence consists of triplets of ones and zeros.¹ The output of the taps is a sequence of correlation scores that can range anywhere from 0 to 108. A score of 108 indicates that every sample is correct. A value of 0 indicates that every sample is

¹ The reason for using three samples per bit period from the limiter/discriminator rather than just one is to help provide a means of accurately locating the ideal bit-sampling time in order to optimize demodulation performance. This is most conveniently done if the number of samples included in the correlation sum is an odd divisor of six (i.e., three).

incorrect. Normally, this latter case would only occur if a ground message were present since the ground message synchronization sequence is just the “opposite” of the ADS-B sequence. These correlation scores are fed at a rate of 6.25 millions scores per second to the threshold detector (see below), and the bits exiting the back end of the delay line are fed at a rate of 6.25 Mbps into the box labeled “memory.”

The **threshold detector** compares the sequence of correlation samples and looks for values at or above a threshold (e.g., 94). In order to find the center (or ideal sampling point) of each bit, the threshold detector will, when confronted with a string of successive threshold crossings, choose the highest one of the group. Having chosen a potential ADS-B burst start point, the threshold detector attaches a marker to the appropriate location in the “memory” (see below), e.g., the first bit sample of the information portion of the message. This process runs continuously.

The **6xN memory** accepts bit samples from the correlator at a rate of 6.25 Msps and arranges them into six separate sequences. Each separate sequence represents a separate string of potential information bits. Those bit strings flagged by the threshold detector are passed to the box called “processing” to determine which comprise real bursts. Note that the receiver has, at this point, no way to determine whether a threshold crossing corresponds to a long or a short ADS-B message. This will be determined in the “processing” box. In the meantime, all ADS-B messages are treated as if they were long ones. At any given time there can be many potential burst starts indicated by pointers in the memory. These can be due to real messages or false alarms that may or may not overlap with other real messages or false alarms. These are all sorted out in the processor.

The **message processing** begins with each potential ADS-B message being read into a first-in-first-out (FIFO) memory based on the locations of the flags set by the threshold detector. As stated previously every message is temporarily assumed to be a long one. Each message, in turn, is then subjected to Reed-Solomon decoding. Those that are successfully decoded are assumed to be valid long ADS-B messages. Those that fail are then subjected to Reed-Solomon decoding assuming they are short ADS-B messages. Those that pass this test are assumed to be valid. Potential messages that fail to decode either way are discarded. Because of the very low undetected error probabilities of the Reed-Solomon codes employed by the ADS-B messages, this method of sorting through the potential message should be extremely reliable.

Note that the ability to process multiple overlapping messages is limited only by the size of the 6xN memory and the ability of the Reed-Solomon decoder to process all potential messages. It appears that neither of these is a real problem. The memory requirement is very small, and the decoder chip in the current UAT prototype can handle many times the maximum required decode rate.

3. Threshold Setting

Given that the length of the sequence is 36 bits, the only free parameter controlling synchronization performance is the threshold value. This is currently set at or about 94. A way to interpret the meaning of this value is to assume the three samples of each bit which are used to determine the correlation value are always the same. If this were the case, then a threshold of 94 would correspond to about 31 correct bits out of 36. In reality, things are not so simple since only the middle of the three samples of a particular bit is “full strength.” Transmitter base-band filtering and a narrow receiver IF filter can muddle things up considerably; however, the assumption that the correlated bits act in groups of three may be a good guide. Nevertheless, any conclusions regarding the threshold values should ultimately be checked using an accurate simulation or testing with actual hardware, if possible.

To assess the “current” performance of UAT vis-à-vis the comparison of synchronization and data, we show the probabilities of synchronization failure (approximate) and decode failure for some of the candidate error correction schemes (RS(27,17), RS(45,33) and RS(85,65)) in figure 2. We see that in the range of interest the synchronization performance is significantly better than that of the data, i.e., in a stationary noise environment almost all failures are expected to be decode failures, not synchronization failures. This will remain true even if slightly stronger codes (e.g., RS(29,17) and RS(47,33)) are used for the ADS-B messages.

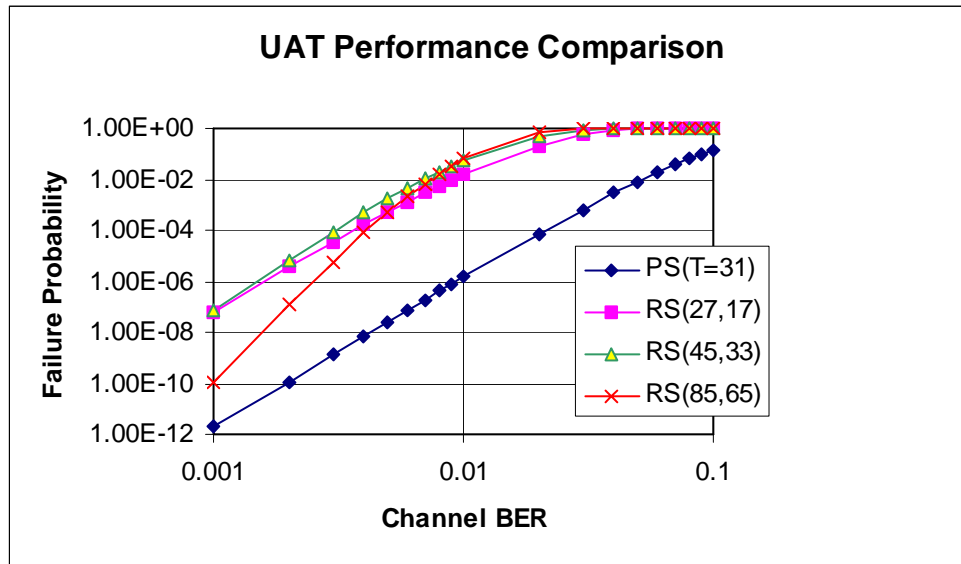


Figure 2. UAT Synchronization and Data Performance

The false alarm rate for the synchronization process will determine the rate at which extra messages are generated by the synchronization process, i.e., those that will need to be later weeded out by the RS decoding process. It is relatively easy to calculate the probability that a random string of ones and zeros will exceed a certain threshold;

however, to get the false alarm *rate*, we need to multiply this probability by the number of *independent* correlation values per second. Due to the assumed narrowness of the receiver IF filter, we will assume that there is one independent sample per bit period (0.96 μ sec). Combining this assumption with the fact that only about 80% of each second is occupied by ADS-B messages gives the estimate of the false alarm rate as

$$FAR = (0.8/0.96 \times 10^{-6}) P_{FA}$$

with

$$P_{FA} = 0.5^{36} \sum_{n=0}^{36-T} \frac{36!}{n!(36-n)!}.$$

T is the threshold. A graph of the false alarm rate versus the threshold value is shown in figure 3.

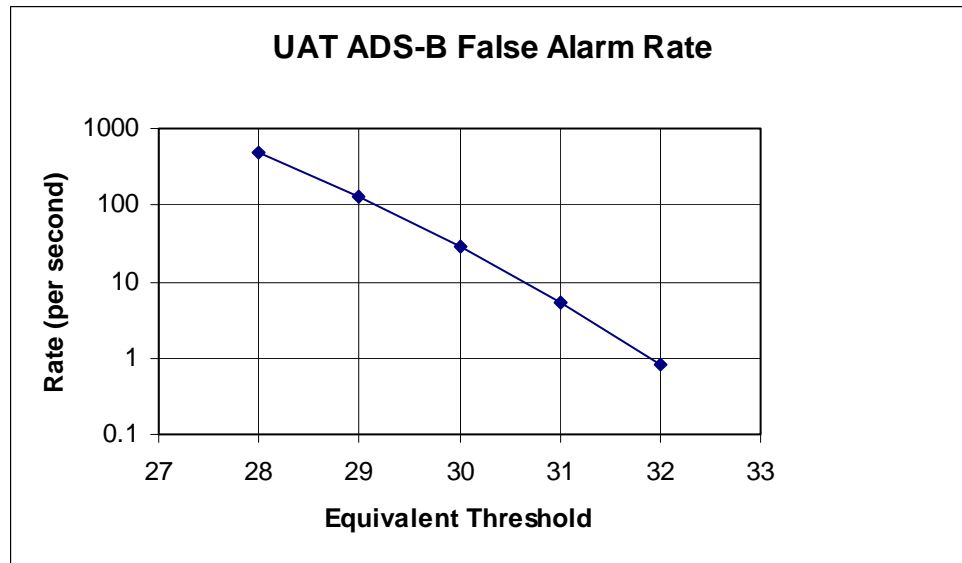


Figure 3. UAT False Alarm Rates

If all our assumptions are correct, then the false alarm rate with the current threshold (i.e., about 31) is about 3 per second. Since the RS decoder must be capable of decoding hundreds of messages per second, a few extras can cause no harm. As a matter of fact, it appears that the equivalent threshold could be lowered a little bit, if desired, to perhaps the equivalent of 29. In terms of the actual threshold values as used in the receiver, this means that the threshold could possibly be lowered from 94 to about 87. Note that lowering the threshold will have the effect of increasing the probability of detection when an actual signal is present.

4. Summary

This paper has described a possible mechanism for allowing a UAT receiver to synchronize to and demodulate multiple overlapping ADS-B signals. This capability is essential for successful operation in environments with many aircraft. The suggested technique does not require extensive resources in terms of memory space or processing speed.

The paper also briefly examines the choice of the threshold value for the synchronization process. Using certain simplifying assumptions, it is shown that the current setting appears to be acceptable, but it could be adjusted somewhat. A prudent approach to choosing the correct threshold value would be to measure the synchronization rate of an actual UAT receiver when no valid signal is present. The threshold could then be set by choosing a value which gives rise to an acceptably small number of false alarms per second.